DCATT PM ACTUATOR CONTROL AND ERROR MINIMIZATION

Updated: 7-29-98

Brendon Perkins
301-286-9080
bperkins@mscmail.gsfc.nasa.gov

INTRODUCTION

This document presents one possible control scheme for minimizing position errors during normal operation of the DCATT testbed. Based on the assessment presented herein, it is expected that position errors on relative positions can be minimized to ±27.5 nm during normal operations of the testbed. It's also expected that absolute positions will have associated position uncertainties on the order of ±77.5 nm. In this case, relative position uncertainty is the uncertainty that is introduced between individual displacements from one position to the next. It could just as well be refered to as displacement uncertainty since it represents nothing more than the uncertainty in the length of travel from one position to the next.

ACTUATOR OPERATION MODES

Motor Stepping (Sub-Stepping)

The most fundamental way in which actuator movements can be implemented is with moter steps (often referred to as substeps). Motor stepping is implemented in an open loop fashion in which an actuator is commanded to move by a specific number of sub-steps. In open loop operations, the actuators can be commanded to substep through the full range of the actuator. Preliminary measurements of the actuators with a Zygo interferometer indicate that substeps produce an average displacement of approximately 2.5 nm with a 20% error.

Encoder Stepping

Encoder steps are really just motor steps performed in a closed loop fashion. The encoder closes the loop by generating two signals with wavelengths of 500 nm that are out of phase by 90 degrees. These signals are essentially XORed together to produce a digital signal of alternating sequences of 0's and 1's which are then used by the controller to determine whether an actuator is within the range of a particular encoder step. The word "range" is the key word since the controller does not receive discrete blips that mark the start and end positions of each encoder step. In other words, the controller does not complete relative movements by moving an actuator to a discrete final position. Instead, it moves the actuator to a final position that falls within a discrete range (an encoder step) which is 125 nm in width.

```
Signal 0                       |<--------------------- 500 nm -------------------->|

     |1111111111111111111111111|                         |1111111111111111111111111|
     |                         |                         |                         |
   <-|-------------------------|-------------------------|-------------------------|->
     |                         |                         |                         |
   11|                         |0000000000000000000000000|                         00

Signal 1          |<-------------------- 500 nm --------------------->|

                  |1111111111111111111111111|                         |11111111111111
                  |                         |                         |
   <--------------|-------------------------|-------------------------|-------------->
                  |                         |                         |
    11111111111111|                         |0000000000000000000000000|


Encoder Steps (XORed Signal)

                            -->|   125 nm   |<--
   <-|------------|------------|------------|------------|------------|------------|->
    0|111111111111|000000000000|111111111111|000000000000|111111111111|000000000000|1
[-]<-|--Step x-3--|--Step x-2--|--Step x-1--|---Step x---|--Step x+1--|--Step x+2--|->[+]

 NOTE: Movement in the negative direction is equivelent to actuator length contraction
       while movement in the positive direction is equivelent to actuator length
       expansion.
```

Nevertheless, the final position can be inferred from off-line, statistical measurements to a much higher level of fidelity as a function of actuator speed. In order to gain a better understanding, consider the following two examples in which an actuator is commanded to move at two different speeds to a destination encoder step. Example A represents what can happen when an actuator is commanded to move at a high rate of speed (TBD) in the closed loop encoder mode. Notice that the actuator may overshoot its destination position before being forced back by the controller. In this case, the position uncertainty could be as great as +125.0 nm from the negative edge of the destination encoder step. However, preliminary measurements of the actuators have shown that this uncertainty can be reduced substantially by operating the actuators at slower speeds.

```
EXAMPLE A: High Actuator Speeds

[-]<-|------------|--S---------|------------|------------|--------E---|------------|->[+]
                     T                                            N
                     A                                            D
                     R                                            |<-----------|
                     T                                                         |
                     |------------------------> high ------------------------->|
```

Now consider Example B. At a sufficiently low rate of speed (TBD), it is expected that an actuator will repeatedly come to rest just inside the edge of the destination encoder step. In this case, the position uncertainty is also expected to be no greater than a single motor step (or 2.9 nm) from the edge of the step. Furthermore, the encoder signal is expected to be highly regular such that the edges of the encoder steps can be used as local position references to minimize error when implementing large actuator displacements. However, it’s important to note that the phase of the encoder signals with respect to the actuators’ end of range positions will not be known. Therefore, absolute actuator positions will never be known to better than ± 62.5 nm.

```
EXAMPLE B: Slow Actuator Speeds

[-]<-|------------|--S---------|------------|------------|E-----------|------------|->[+]
                     T                                    N
                     A                                    D
                     R                                    |
                     T                                    |
                     |--------------> Slow -------------->|
```

Closed Loop vs Open Loop Operations

In the ensuing text, the phrases “closed loop” and “open loop” are used only to refer to the state of the low level, hardware controller. With encoder stepping, the low level, hardware controller monitors the encoder signal to close the loop and ensure that the actuator is moved to a specific destination encoder step. On the other hand, motor stepping is considered to be an open loop operation since the low level, hardware controller is not employed to force the actuator to a particular position. Nevertheless, the encoder signal can still be monitored during open loop operations to manually close the loop. In fact, this is done in operations III and IV of the next section.

PROPOSED ACTUATOR CONTROL SCHEME

Based on the information of the preceeding sections, it's possible to propose a six operation control scheme that involves a rapid displacement over multiple encoder steps followed by three slow displacements within the final encoder step to achieve the high position accuracy that is required without sacrificing the speed that will be desired when moving over multiple encoder steps. This control scheme would be implemented whenever the OTA controller accepts a command to implement a relative displacement specified in engineering units.

Operation I

Retrieve the current absolute position of the actuator. This position will be specified in encoder steps with a fractional component. From this position value, the OTA controller will calculate the current relative position and relative position uncertainty with respect to the local position reference (the negative edge of the current encoder step). The OTA controller will then calculate the number of encoder steps that must be moved to reach the encoder step containing the destination ("END") position.

```
    +|-          +|-          +|-          +|-          +|-          +|-          +|-
[-]<-|------------|--S---------|------------|------------|-------E----|------------|->[+]
                  ^  T                                           N
                  |  A                                           D
                  |  R
                  |  T
     Position     |
     Local Ref----/
```

Operation II

The OTA controller will then command the actuator to move at a high rate of speed using closed loop encoder stepping to reach the encoder step that contains the destination position. The final position of this operation is the first of three midpoint positions that must be reached before arriving at the destination position. This first midpoint position will have an uncertainty of up to +125 nm from the negative edge of the encoder step since it can reside anywhere within the full range of the step.

```
                                                                  0.0 nm ----> 125 nm
    +|-          +|-          +|-          +|-          +|-          +|-          +|-
[-]<-|------------|--S---------|------------|------------|---x---E----|------------|->[+]
                     T                                       |   N
                     A                                       |   D
                     R                                       |
                     T                                       |
                     |-------------------------------------->|
```

Operation III

In order to reduce the position uncertainty introduced by the previous operation, The OTA controller will then command the actuator to move towards the edge of the encoder step that is closest to the destination position. More specifically, the actuator will be commanded to move at a slow rate of speed using open loop motor stepping while monitoring the encoder signal until it switches states (indicating that the next encoder step has been reached). The actuator will then be positioned just outside the destination encoder step with a position uncertainty on the order of a single motor step (or 2.9 nm)

```
    +|-          +|-          +|-          +|-          +|-          +|-          +|-
[-]<-|------------|--S---------|------------|------------|---x---E----|x-----------|->[+]
                     T                                       |   N     |
                     A                                       |   D     |
                     R                                       |         |
                     T                                       |         |
                     |-------------------------------------->|-------->|
```

Operation IV

Although the position uncertainty has been reduced substantially, it is not desireable to leave the actuator positioned outside the final encoder step. Therefore, the OTA controller will command the actuator to move back into the destination encoder step. More specifically, the actuator will be commanded to move at a slow rate of speed using open loop motor stepping until the encoder signal switches states once again (indicating that the destination encoder step has been reached). This ensures that the final actuator position will reside within the destination encoder step which is, otherwise, not gauranteed. To clarify this statement, consider the case in which the edge of the encoder step is also the final destination position. It is reasonable to expect that no subsequent motor stepping would be commanded after the completion of Operation III. In that event, the actuator would be positioned outside the range of the destination encoder step which is not desireable since consistent positioning of the actuator is expected between individual activations of closed loop control. Therefore, implementation of this operation prevents a potential position ambiguity from occuring. At this point, the position uncertainty will still be on the order of a single motor step (or 2.9 nm).

```
    +|-          +|-          +|-          +|-          +|-          +|-          +|-
[-]<-|------------|--S---------|------------|------------|---x---E---x|x-----------|->[+]
                     T                                       |   N   | |
                     A                                       |   D   | |
                     R                                       |       | |
                     T                                       |       |<|
                     |-------------------------------------->|-------->|
```

Operation V

The actuator will now be moved to its final position at a slow rate of speed using open loop motor stepping. In other words, the OTA controller will command the actuator to move towards the center of the encoder step by a specific number of motor steps to place the actuator at its final position. It's important to note that the final position of the actuator cannot be measured directly. However off-line measurements of the actuators will provide statistical data for determining the number of motor steps that are required for reaching the final position. These measurements will also provide data for calculating the associated position uncertainty. It is obvious that the worste case position uncertainty will occur whenever an actuator must be moved to the center of the encoder step which is equivelent to 62.5 nm (125 nm / 2) of motor step travel. Based on preliminary measurements of the actuators, the contribution of this operation to the overall position uncertainty is expected to be ±(20%)*(62.5 nm) = ±12.5 nm.

```
    +|-          +|-          +|-          +|-          +|-          +|-          +|-
[-]<-|------------|--S---------|------------|------------|---x---E---x|x-----------|->[+]
                     T                                       |   N   | |
                     A                                       |   D   | |
                     R                                       |   |   | |
                     T                                       |   |<--|<|
                     |-------------------------------------->|-------->|
```

Operation VI

The final operation consists of nothing more than logging the final position of the actuator as well as calculating the corresponding relative and absolute position uncertainties.

ACTUATOR POSITION UNCERTAINTY

Relative Position Uncertainty

The relative position uncertainty is simply the sum of three errors: (1) the position uncertainty of the initial position with respect to the local position reference, (2) the position uncertainty that is associated with moving to the local position reference of the destination encoder step, and (3) the additional position uncertainty that is introduced when moving towards the center of the final encoder step to reach the destination position. These are the position uncertainties associated with Operations I, IV, and V, and the worste case is:

Maximum Relative Position Error = ERR_op_i + ERR_op_iv + ERR_op_v

= ±12.5 nm ± 2.5 nm ± 12.5 nm

= ±27.5 nm

Keep in mind that the maximum error contribution for Operation I is the same as that for operation Operation V.

Absolute Position Uncertainty

The absolute position uncertainty also has three error contributions as well. They are: (1) the phase of the encoder signal with respect to the actuator end of range position, (2) the position uncertainty that is associated with finding local position references, and (3) the additional position uncertainty that is introduced when moving towards the center of the final encoder step to reach the destination position.

Maximum Absolute Position Error = ERR_ecd_signal + ERR_op_iv + ERR_op_v

= ± 62.5 nm ± 2.5 nm ± 12.5 nm

= ± 77.5 nm

As you can see, the error in absolute position is quite substantial. In this case, the major contribution is a lack of knowledge of the encoder signal phase when moving away from the end of range position for the actuator. It’s important to note that this end of range position is currently the only reference that is available for initializing an actuator to its center of range position during normal operations of the testbed. It is not likely that the error in absolute position will be reduced unless another reference to absolute position can be provided.

SAMPLE MATLAB CODE

Now that we have a proposed control scheme for the actuators, it's possible to identify how position uncertainties arise during normal operations of the actuators. The following matlab code embodies the control logic presented in the previous sections and is a first stab at generating code that can be used in completing actuator sensitivity studies.

```
[REL_ERR, ABS_ERR] = ActuatorControl (Displace, seg, act)

    % The following variables are constants to be determined from off-line measurements
    % of the actuators. All values are given in [mm].

    MTR_SIZE = 2.9e-7;   % Mean – Motor Step Size
    ECD_SIZE = 125e-7;   % Mean - Encoder Step Size
    ECD_ERR  = MTR_SIZE; % Encoder Stepping Position Uncertainty
    RNG_SIZE = 6;        % Absolute Actuator Range
    RNG_ERR  = ECD_SIZE; % Absolute Actuator Range Error

    % Additional variables used in this function include

    % Displace – Commanded displacement relative to the initial actuator position.
    % DISP_ref – Commanded displacement relative to the local reference position of the
    %            encoder step in which the actuator initially resides.
    % Distance -
    % ECD_STPS – Used to track encoder steps
    % MTR_STPS - Used to track motor steps

    LOAD ACT_DATA;

    % ACT_DATA Contains:
    %
    % REL_POS (seg,act) – An array that holds the relative position of each actuator off its
    %                     local position reference.
    % dREL_POS(seg,act) – An array that holds the uncertainty in the relative positions
    %                     provided in the REL_POS array.
    % REL_ERR(seg,act)  - An array that holds the uncertainty of the last displacement that
    %                     was implemented for each actuator in the system.
    % ABS_ERR(seg,act)  - An array that holds the uncertainty in the current absolute
    %                     position of each actuator in the system.

    % Calculate the number of encoder steps to be moved

    DISP_ref = REL_POS(seg,act) + Displace;
    ECD_STPS = floor ( DISP_ref / ECD_SIZE );

    IF (ECD_STPS .NE. 0.0) THEN

       % Command the actuator to move at a high rate of speed using closed loop encoder
       % stepping. Move the actuator by the number of encoder steps specified in ECD_STPS.
       % Negative values in ECD_STPS correspond to actuator contractions while positive
       % values correspond to actuator extensions.

       % LOW LEVEL ACTUATOR COMMAND(S) GO HERE

       % The actuator position uncertainty will now rise to +125 nm from the negative edge
       % of the destination encoder step (the local position reference).

       ECD_ERR = ECD_SIZE;

       % Calculate the distance of the destination position from the negative edge of the
       % current encoder step (the local reference position).

       REL_POS = DISP_ref - ECD_STPS * ECD_SIZE;

       % Determine which local position reference is closest to the destination position.
       % If it's not the local position reference of the current encoder step, it will be
       % the local position reference of the encoder step on the positive side of the
       % current encoder step. Move to the closest reference position to minimize position
       % uncertainty and then move to the destination position.
```

```
IF (REL_POS .LT. 0.0) THEN

   IF (REL_POS .LT. –75) THEN  //  -125 nm < REL_POS < -75 nm

      % Command the actuator to move at a slow rate of speed in the negative
      % direction using open loop motor stepping while continuously reading the
      % encoder signal until the signal indicates that the encoder step’s negative
      % edge (the local position reference of the next encoder step) has been
      % crossed.

      % LOW LEVEL ACTUATOR COMMAND(S) GO HERE

      % At this point the actuator will be positioned just outside the destination
      % encoder step and must be pushed back in. Command the actuator to move at a
      % slow rate of speed in the positive direction using open loop motor stepping
      % while continuously reading the encoder signal until the signal indicates that
      % the actuator is positioned within the destination encoder step once again.

      % LOW LEVEL ACTUATOR COMMAND(S) GO HERE

      % Now command the actuator to move at a slow rate of speed in the positive
      % direction using open loop motor stepping using off-line statistical
      % measurements to calculate the required number of motor steps and the
      % associated position uncertainty from the local reference.

      MTR_STPS = ABS ( (-125–REL_POS) / MTR_SIZE );
      MTR_ERR  = 0.20*MTR_STPS;

      % LOW LEVEL ACTUATOR COMMAND(S) GO HERE

   ELSE                         // -75 nm <= REL_POS < 0 nm

      % Command the actuator to move at a slow rate of speed in the positive
      % direction using open loop motor stepping while continuously reading the
      % encoder signal until the signal indicates that the encoder step’s positive
      % edge (the local position reference of the next encoder step) has been
      % crossed.

      % LOW LEVEL ACTUATOR COMMAND(S) GO HERE

      % At this point the actuator will be positioned just outside the destination
      % encoder step and must be pushed back in. Command the actuator to move at a
      % slow rate of speed in the negative direction using open loop motor stepping
      % while continuously reading the encoder signal until the signal indicates that
      % the actuator is positioned within the destination encoder step once again.

      % LOW LEVEL ACTUATOR COMMAND(S) GO HERE

      % Now command the actuator to move at a slow rate of speed in the negative
      % direction using open loop motor stepping using off-line statistical
      % measurements to calculate the required number of motor steps and the
      % associated position uncertainty from the local reference.

      MTR_STPS = ABS ( REL_POS / MTR_SIZE );
      MTR_ERR  = 0.20*MTR_STPS;

      % LOW LEVEL ACTUATOR COMMAND(S) GO HERE

   ENDIF

ELSE

   IF (REL_POS .LT. 75) THEN   % 0 nm <= REL_POS < 75 nm

      % Command the actuator to move at a slow rate of speed in the negative
      % direction using open loop motor stepping while continuously reading the
      % encoder signal until the signal indicates that the encoder step’s positive
      % edge (the local position reference of the next encoder step) has been
      % crossed.

      % LOW LEVEL ACTUATOR COMMEND(S) GO HERE
```

```
         % At this point the actuator will be positioned just outside the destination
         % encoder step and must be pushed back in. Command the actuator to move at a
         % slow rate of speed in the positive direction using open loop motor stepping
         % while continuously reading the encoder signal until the signal indicates that
         % the actuator is positioned within the destination encoder step once again.

         % LOW LEVEL ACTUATOR COMMEND(S) GO HERE

         % Now command the actuator to move at a slow rate of speed in the positive
         % direction using open loop motor stepping using off-line statistical
         % measurements to calculate the required number of motor steps and the
         % associated position uncertainty from the reference position.

         MTR_STPS = ABS ( REL_POS / MTR_SIZE );
         MTR_ERR  = 0.20*MTR_STPS;

         % LOW LEVEL ACTUATOR COMMAND(S) GO HERE

      ELSE                         % 75 nm <= REL_POS < 125 nm

         % Command the actuator to move at a slow rate of speed in the positive
         % direction using open loop motor stepping while continuously reading the
         % encoder signal until the signal indicates that the encoder step’s positive
         % edge (the local position reference of the next encoder step) has been
         % crossed.

         % LOW LEVEL ACTUATOR COMMAND(S) GO HERE

         % At this point the actuator will be positioned just outside the destination
         % encoder step and must be pushed back in. Command the actuator to move at a
         % slow rate of speed in the negative direction using open loop motor stepping
         % while continuously reading the encoder signal until the signal indicates that
         % the actuator is positioned within the destination encoder step once again.

         % LOW LEVEL ACTUATOR COMMEND(S) GO HERE

         % Now command the actuator to move at a slow rate of speed in the negative
         % direction using open loop motor stepping using off-line statistical
         % measurements to calculate the required number of motor steps and the
         % associated position uncertainty from the reference position.

         MTR_STPS = ABS ( (125 – REL_POS) / MTR_SIZE );
         MTR_ERR  = 0.20*MTR_STPS;

         % LOW LEVEL ACTUATOR COMMAND(S) GO HERE

      ENDIF

% Calculate the Relative and Absolute position Errors

   REL_ERR = dREL_POS + ECD_ERR + MTR_ERR;

   ABS_ERR = LMT_ERR  + ECD_ERR + MTR_ERR;

  dREL_POS = MTR_ERR;
```